UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 09/855,199 | 05/14/2001 | Vijaya Raghavan | 04899-044001 | 8175 |

74321 7590 11/30/2007

LAHIVE & COCKFIELD, LLP/THE MATHWORKS
One Post Office Square
Boston, MA 02109-2127

| EXAMINER | |
|---|---|
| ALHIJA, SAIF A | |
| ART UNIT | PAPER NUMBER |
| 2128 | |

| MAIL DATE | DELIVERY MODE |
|---|---|
| 11/30/2007 | PAPER |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

| Office Action Summary | Application No. | Applicant(s) | |
|---|---|---|---|
| | 09/855,199 | RAGHAVAN ET AL. | |
| | Examiner | Art Unit | |
| | Saif A. Alhija | 2128 | |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *01 October 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1-6,12-18 and 20-78* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-6, 12-18 and 20-78* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *24 August 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

## *DETAILED ACTION*

**1.** Claims 1-6, 12-18, and 20-78 have been presented for examination.

Claims 7-11 and 19 have been cancelled.

### *Response to Arguments*

**2.** A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 1 October 2007 has been entered.

i) Examiner thanks the Applicant for agreeing to the change of title. An amendment to that effect is still required.

ii) Applicant argues that the reference does not disclose "representing the at least one function graphically and separately from the at least one state or transition..." First it is unclear what is meant by separately in the context of Applicants claim and arguments. How is the separation aspect a novel feature or rather how is the separation not disclosed by Kodosky. Second, Kodosky discloses in paragraphs 9, 10, and 12 reproduced below the concept and implementation of a graphical program which contains functions.

[0009] The method disclosed in Kodosky et al allows a user to construct a diagram using a block diagram editor. The block diagram may include a plurality of interconnected icons such that the diagram created graphically displays a procedure or method for accomplishing a certain result, such as manipulating one or more input variables and/or producing one or more output variables. The diagram may have one or more of data flow, control flow and/or execution flow representations. In response to the user constructing a diagram or graphical program using the block diagram editor, data structures may be automatically constructed which characterize an execution procedure which corresponds to the displayed procedure. The graphical program may be compiled or interpreted by a computer.

[0010] Therefore, Kodosky et al teaches a graphical programming environment wherein a user places or manipulates icons and interconnects or "wires up" the icons in a block diagram using a block diagram editor to create a graphical "program." A graphical program for measuring, controlling, or modeling devices, such as instruments, processes or industrial automation hardware, or for modeling or simulating devices, may be referred to as a virtual instrument (VI). Thus, a user can create a computer program solely by using a graphically based programming environment. This graphically based programming environment may be used for creating virtual instrumentation systems, modeling processes, control, simulation and numerical analysis, as well as for any type of general programming.

[0012] During creation of the block diagram portion of the graphical program, the user may select various function nodes or icons that accomplish his desired result and connect the function nodes together. For example, the function nodes may be connected in one or more of a data flow, control flow, and/or execution flow format. The function nodes may also be connected in a "signal flow" format, which is a subset of data flow. The function nodes may be connected between the terminals of the various user interface elements, e.g., between the respective controls and indicators. Thus the user may create or assemble a graphical program, referred to as a block diagram, graphically representing the desired process. The assembled graphical program may be represented in the memory of the computer system as data structures. The assembled graphical program, i.e., these data structures, may then be compiled or interpreted to produce machine language that accomplishes the desired method or process as shown in the block diagram.

iii) Applicant argues that the reference does not disclose "modifying the at least one function through graphical diagramming." This can be seen in paragraph 10 above.

iv) Applicant argues that the reference does not disclose a function call. This can be seen in paragraph 12 above.

v) Applicant argues that the reference does not disclose a function flow diagram. Applicants have reproduced the cited section of the reference however no arguments have been presented which show that the data flow, control flow, and execution flow recited in the reference do not anticipate the function flow as claimed. Applicants have merely stated that they are not equivalent however no specifics have been addressed.

vi) Applicant argues that the reference does not disclose "means for hiding the display of the function flow diagram based upon user input." Applicants have merely reiterated that the reference does not teach this feature and have not addressed the Examiners argument which was presented in the previous office action and is subsequently repeated. This feature can be simply seen in the graphical environment provided by the reference in which certain aspects of the graphical program or state diagram are not selected and therefore not seen in the GUI. This is not a novel feature and is akin to closing a window in a GUI.

vii) Applicant argues that the reference does not disclose a function prototype. This can be seen in Paragraphs 177-178, reproduced below, with respect to the integration of text-based programs such as C++, Java, etc as well as the client graphical program's integration with the text-based code which inherently include function prototypes. See also paragraph 64.

> **[0177]** As noted above, the client program **502** may be any of various types of programs. For example, the client program **502** may be a graphical program. The client program **502** may also be a text-based program such as a C++ program, a Visual Basic program, a Java program, etc., or any combination of these or other languages. The client program **502** may execute independently or may execute within an execution subsystem of an application development environment.
>
> **[0178]** The client program **502** may call the API **504** in any of various ways. For example, a client graphical program may include graphical nodes corresponding to the API **504**. A client graphical program may also interface with text-based code which calls the API **504**. The client program **502** may also call the API **504** in various other ways. For example, the server program **506** may expose a component such as an ActiveX component, CORBA component, JavaBeans component, etc., and the client program **502** may obtain a reference to the object to invoke functions or methods of the API **504**. The API **504** may also be integrated with the language or development environment of the client program **502**, e.g. as a library.

viii) Applicant argues that the reference does not disclose a function having two or more graphical elements. This can be seen in paragraphs 10 and 12 above as well as at least Figure 8 which contains multiple graphical elements.

ix) The Examiner respectfully appreciates Applicants attempt to further clarify the claimed invention with an amendment to the claims. However, when taken in the broadest most reasonable interpretation there does not appear to be a patentable distinction between the presented claims and the provided reference. The Examiner respectfully encourages Applicants to provide further specificity to overcome the broad nature the claims since the Examiner must consider patentable distinction when formulating a decision with respect to the patentability of the presented claims.

x) Examiner has cited particular columns and line numbers in the references applied to the claims for the convenience of the applicant. Although the specified citations are representative of the teachings of the art and are applied to specific limitations within the individual claim, other passages and figures may apply as well. It is respectfully requested from the applicant in preparing responses, to fully consider the references in their entirety as

potentially teaching all or part of the claimed invention, as well as the context of the passage as taught by the prior art or disclosed by the Examiner.

xi) The Examiner respectfully requests, in the event the Applicants choose to amend or add new claims, that such claims and their limitations be directly mapped to the specification, which provides support for the subject matter. This will assist in expediting compact prosecution.

xii) Further, the Examiner respectfully encourages Applicants to direct the specificity of their response with regards to this office action to the broadest reasonable interpretation of the claims as presented. This will avoid issues that would delay prosecution such as limitations not explicitly presented in the claims, intended use statements that carry no patentable weight, mere allegations of patentability, and novelty that is not clearly expressed.

***Claim Rejections - 35 USC § 102***

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

3. **Claim(s) 1-6, 12-18, 20-78** are rejected under 35 U.S.C. 102(e) as being clearly anticipated by **Kodosky et al. "System and Method for Programmatically Generating a Graphical Program in Response to a State Diagram" U.S. Patent Application Publication # 2002/0083413 A1.**

**Regarding Claim 1:**

**Kodosky et al. discloses** a computer-implemented method comprising:

providing a graphical user interface for defining at least one function to be used in a graphical representation of a finite state machine where the graphical representation is an executable model of the finite state machine and includes at least one state or transition; **(Page 14, Paragraph 165, Lines 1-5. Figure 19)**

representing the at least one function graphically and separately from the at least one state or transition in the graphical representation of the finite state machine, wherein the function that is represented graphically is a function defined in a graphical language; and **(Page 14, Paragraph 165, Lines 1-5. Figure 19)**

calling the function that is represented graphically from within the finite state machine. **(Page 15, Paragraph 166, Lines 11-15)**

**Regarding Claim 2:**

**Kodosky et al. discloses** the method of claim 1 wherein defining the at least one function further comprises using a function block. **(Page 14, Paragraph 165, Lines 7-9. Figure 19)**

**Regarding Claim 3:**

**Kodosky et al. discloses** the method of claim 2 wherein defining the at least one function further comprises using a function prototype. **(Page 2, Paragraph 11, Lines 1-6)**

**Regarding Claim 4:**

**Kodosky et al. discloses** the method of claim 1 wherein the defining step further comprises using a function flow diagram. **(Page 1, Paragraph 9, Lines 7-9)**

**Regarding Claim 5:** a computer-implemented method, said method comprising

providing a graphical user interface for defining at least one function to be used in a graphical representation of a finite state machine where the graphical representation is an executable model of the finite state machine and includes at least one state or transition **(Page 14, Paragraph 165, Lines 1-5. Figure 19)**

representing the at least one function graphically and separately from the at least one state or transition in the graphical representation of the finite state machine, wherein the function is represented graphically as a diagram comprising graphical elements; and **(Page 14, Paragraph 165, Lines 1-5. Figure 19)**

calling the function that is represented graphically from within the finite state machine. **(Page 15, Paragraph 166, Lines 11-15)**

**Regarding Claim 6:**

**Kodosky et al. discloses** the method of claim 1 further comprising modifying the at least one function through graphical diagramming. **(Figure 8)**

**Regarding Claim 12:**

**Kodosky et al. discloses** a computer program product, stored in a computer readable medium, comprising instructions to cause a computer to:

receive user input defining at least one graphical function for use in a finite state machine; **(Page 14, Paragraph 165, Lines 1-5. Figure 19)**

use the at least one graphical function in a simulation of a system represented by the finite state machine, wherein the instructions to use the at least one graphical function further comprise instructions to call the at least one graphical function from at least one state or transition in the finite state machine. **(Page 14, Paragraph 165, Lines 1-5. Figure 19)**

the at least one graphical function being represented graphically and separately from the at least one state or transition in the finite state machine. **(Page 14, Paragraph 165, Lines 1-5. Figure 19)**

**Regarding Claim 13:**

**Kodosky et al. discloses** the computer program product of claim 12 wherein the user input defining the at least one graphical function is entered into a function block. **(Page 1, Paragraph 9, Lines 1-2)**

**Regarding Claim 14:**

**Kodosky et al. discloses** the computer program product of claim 12 wherein the user input defining the at least one graphical function includes a function prototype. **(Page 2, Paragraph 11, Lines 1-6)**

**Regarding Claim 15:**

**Kodosky et al. discloses** the computer program product of claim 12 wherein the user input comprises a function flow diagram. **(Page 1, Paragraph 9, Lines 7-9)**

**Regarding Claim 16:**

**Kodosky et al. discloses** the computer program product of claim 15 wherein the function flow diagram is a comprised of graphical elements. **(Figure 8)**

**Regarding Claim 17:**

**Kodosky et al. discloses** a system for modeling finite state machines said system comprising:

a computer comprising a graphical user interface, memory, storage, and at least one input device; **(Page 6, Paragraph 63, Lines 1-4)**

means to receive user input to define at least one graphical function; **(Page 6, Paragraph 63, Lines 1-8)**

means to represent the graphical function as an executable state flow diagram; **(Page 2, Paragraph 16, Lines 4-9)**

means to call the graphical function from at least one finite state machine in a simulation of the at least one finite state machine the finite state machine including at least one state or transition the graphical function being represented graphically and separately from the at least one state or transition in the finite state machine. **(Page 15, Paragraph 166, Lines 13-20)**

**Regarding Claim 18:**

**Kodosky et al. discloses** the system of claim 17 wherein the user input to define the at least one graphical function is entered into a function block. **(Page 1, Paragraph 9, Lines 1-2)**

**Regarding Claim 20:**

**Kodosky et al. discloses** the system of claim 17 wherein the user input defining the at least one graphical function includes a function prototype. **(Page 2, Paragraph 11, Lines 1-6)**

**Regarding Claim 21:**

**Kodosky et al. discloses** the system of claim 17 wherein the user input comprises a function flow diagram. **(Page 1, Paragraph 9, Lines 7-9)**

**Regarding Claim 22:**

**Kodosky et al. discloses** the system of claim 21 wherein the function flow diagram is comprised of graphical elements. **(Figure 8)**

**Regarding Claim 23:**

**Kodosky et al. discloses** the system of claim 21 further comprising means for hiding the display of the function flow diagram based upon user input. **(Page 1, Paragraph 9, Lines 7-9)**

**Regarding Claim 24:**

**Kodosky et al. discloses** a method of operating a data processing system having a graphical user interface said method comprising:

creating a graphical representation of a finite state machine and a graphical representation of a function for use in the graphical representation of the finite state machine the graphical representation of the finite state machine including at least one state or transition the graphical representation of the function being represented graphically and separately from the at least one state or transition in the graphical representation of the finite state machine, **(Page 1, Paragraph 9, Lines 9-14)**

simulating a system represented by the finite state machine wherein the graphical representation is an executable model of the system; and **(Page 1, Paragraph 10, Lines 10-13)**

calling the function from the executable model of the system during the act of simulating the system represented by the finite state machine **(Page 14, Paragraph 165, Lines 1-5. Paragraphs 168-172. Figure 19)**

**Regarding Claim 25:**

**Kodosky et al. discloses** the method of claim 24 wherein the graphical representation of the function comprises a function prototype. **(Page 2, Paragraph 11, Lines 1-6)**

**Regarding Claim 26:**

**Kodosky et al. discloses** the method of claim 25 wherein the function prototype defines a textual format for invoking the function. **(Paragraph 132 Lines 1-5. Figure 8)**

**Regarding Claim 27:**

**Kodosky et al. discloses** the method of claim 26 wherein the graphical representation of the finite state machine includes at least one invocation of the function using the defined textual format. **(Page 12, Paragraph 132 Lines 1-5. Figure 8)**

**Regarding Claim 28:**

**Kodosky et al. discloses** the method of claim 24 further comprising shadowing a function, wherein shadowing comprising using in a function invocation a function definition closest to a point of invocation of the function in a state diagram hierarchy. **(Page 3, Paragraph 20, Lines 8-13; Creators priority order can allow for closest function definition to execute.)**

**Regarding Claim 29**

**Kodosky et al. discloses** the method of claim 24 wherein the function is exportable by a state chart and may be invoked anywhere in the finite state machine in which the chart appears, including other charts that define the finite state machine. **(Page 3, Paragraph 26, Lines 4-10. Page 9, Paragraph 100, Lines 1-5)**

**Regarding Claim 30:**

**Kodosky et al. discloses** the method of claim 24 wherein simulating the system represented by the finite state machine further comprises computer code generation. **(Page 12, Paragraph 133, Lines 1-4)**

**Regarding Claim 31:**

**Kodosky et al. discloses** the method of claim 24

wherein the graphical representation of the function comprises a function prototype defining a textual format for invoking the function; **(Page 12, Paragraph 132, Lines 1-5. Figure 8)**

and wherein the graphical representation of the finite state machine includes an invocation of the function using the defined textual format. **(Page 12, Paragraph 132, Lines 1-5. Figure 8)**

**Regarding Claim 32:**

**Kodosky et al. discloses** a computer readable medium having encoded thereon

instructions for causing a computer system to receive through a graphical user interface graphical representation of a finite state machine and a graphical representation of at least one function for use in the graphical representation of the finite state machine the graphical representation of the finite state machine including at least one state or transition, the graphical representation of the function being represented graphically and separately from the at least one state or transition in the graphical representation of the finite state machine; and **(Page 1, Paragraph 9, Lines 9-14)**

instructions for simulating a system represented by the finite state machine where the graphical representation is an executable model of the system; and **(Page 1, Paragraph 10, Lines 10-13)**

instructions for calling the function from at least one place in the executable model during the system simulation. **(Page 14, Paragraph 165, Lines 1-5. Paragraphs 168-172. Figure 19)**

**Regarding Claim 33:**

**Kodosky et al. discloses** the computer readable medium of claim 32,

wherein the graphical representation of the function comprises a function prototype defining a textual format for invoking the function; **(Page 2, Paragraph 11, Lines 1-6)**

and wherein the graphical representation of the finite state machine includes an invocation of the function using the define textual format. **(Page 12, Paragraph 132 Lines 1-5. Figure 8)**

**Regarding Claim 34:**

**Kodosky et al. discloses** in an electronic device, a method of graphically representing an event-driven system, said method comprising:

Providing one or more block components representing one or more states in an executable model; **(Page 1, Paragraph 9, Lines 1-3)**

Providing one or more transition components representing transitions between the one or more block states; **(Page 2, Paragraph 16, Lines 1-4)** and

Providing a function, said function comprising at least two graphical components and being referenced by at least one the states or at least one of the transitions to call the function at the at least one of the states or the at least one of the transitions the function being represented graphically and separately from the at least one of the state or the at least one of the transition in the executable model. **(Page 2, Paragraph 17, Lines 1-4)**

**Regarding Claim 35:**

**Kodosky et al. discloses** the method of claim 34, wherein the function accepts at least one argument and returns at least one result. **(Page 1, Paragraph 9, Lines 1-4)**

**Regarding Claim 36:**

**Kodosky et al. discloses** the method of claim 34, further comprising invoking the function at a second one of the one or more transition components or one or more block components. **(Page 12, Paragraph 132 Lines 1-5. Figure 8. Page 1, Paragraph 10, Lines 1-5)**

**Regarding Claim 37:**

**Kodosky et al. discloses** the method of claim 34 further comprising specifying data properties of the function. **(Page 1, Paragraph 9, Lines 7-9)**

**Regarding Claim 38:**

**Kodosky et al. discloses** the method of claim 34 further comprising associating a data item with the function. **(Page 1, Paragraph 9, Lines 7-9. Page 2, Paragraph 11, Lines 2-7)**

**Regarding Claim 39:**

**Kodosky et al. discloses** the method of claim 34, wherein the function comprises a graphical function. **(Page 6, Paragraph 63, Lines 1-8)**

**Regarding Claim 40:**

**Kodosky et al. discloses** the method of claim 34, wherein the function has a plurality of configurable properties. **(Page 1, Paragraph 10, Lines 1-5)**

**Regarding Claim 41:**

**Kodosky et al. discloses** the method of claim 34, wherein the function defines a textual format for invoking the function. **(Page 12, Paragraph 132 Lines 1-5. Figure 8)**

**Regarding Claim 42:**

**Kodosky et al. discloses** the method of claim 34, further comprising providing a shadowing function, wherein shadowing comprises using in a function invocation a function definition proximally closest to a point of invocation of the function in a state diagram hierarchy. **(Page 3, Paragraph 20, Lines 8-13; Creators priority order can allow for closest function definition to execute.)**

**Regarding Claim 43:**

**Kodosky et al. discloses** in a graphical representation environment, a system for graphically representing an event-driven system, said system comprising:

One or more bock components representing one or more states in an executable model; **(Page 1, Paragraph 9, Lines 1-3)**

One or more transition components representing transitions between the one or more block components representing the states; **(Page 2, Paragraph 16, Lines 1-4) and**

A component representing a graphical function and referenced by at least one of the states or at least one of the transitions to call the function at one of the states or one of the transition the function being represented graphically and separately from the at least one of the states or the at least one of the transitions in the executable model. **(Page 2, Paragraph 17, Lines 1-4)**

**Regarding Claim 44:**

**Kodosky et al. discloses** the system of claim 43, wherein the function accepts at least one argument and returns at least one result. **(Page 1, Paragraph 9, Lines 1-4)**

**Regarding Claim 45:**

**Kodosky et al. discloses** the system of claim 43, wherein at least a subset of the one or more block components representing the states and the one or more transition components can invoke the function. **(Page 12, Paragraph 132 Lines 1-5. Figure 8. Page 1, Paragraph 10, Lines 1-5)**

**Regarding Claim 46:**

**Kodosky et al. discloses** the system of claim 43, further comprising means for specifying data properties of the function. **(Page 1, Paragraph 9, Lines 7-9)**

**Regarding Claim 47:**

**Kodosky et al. discloses** the system of claim 43, further comprising means for associating a data item with the function. **(Page 1, Paragraph 9, Lines 7-9. Page 2, Paragraph 11, Lines 2-7)**

**Regarding Claim 48:**

**Kodosky et al. discloses** the system of claim 43, wherein the component representing the function is referenced by one more of: at least one of the states or at least on of the transitions. **(Page 6, Paragraph 63, Lines 1-8)**

**Regarding Claim 49:**

**Kodosky et al. discloses** the system of claim 43, wherein the function has a plurality of configurable properties. **(Page 1, Paragraph 10, Lines 1-5)**

**Regarding Claim 50:**

**Kodosky et al. discloses** the system of claim 43, wherein the function defines a textual format for invoking the function. **(Page 12, Paragraph 132 Lines 1-5. Figure 8)**

**Regarding Claim 51:**

**Kodosky et al. discloses** the system of claim 43, further comprising means for providing a shadowing function, wherein shadowing comprises using in a function invocation a function definition proximally closest to a point of invocation of the function in a state diagram hierarchy. **(Page 3, Paragraph 20, Lines 8-13; Creators priority order can allow for closest function definition to execute.)**

**Regarding Claim 52:**

**Kodosky et al. discloses** a medium for use in a graphical representation environment on an electronic device, the medium holding instructions executable using the electronic device for graphically representing an event-driven system, said instructions comprising instructions of:

Providing one or more block components representing one or more states in an executable model; **(Page 1, Paragraph 9, Lines 1-3)**

Providing one or more transition components representing transitions between the one or more block components representing the states; **(Page 2, Paragraph 16, Lines 1-4)** and

Providing a block component representing a graphical function and reference by at least one of the states or at least one of the transitions to call the function at one of the states or one of the transitions during execution of the event-driven system, the function being represented graphically and separately from the at least one of the state or the at least one of the transitions in the executable model. **(Page 2, Paragraph 17, Lines 1-4)**

**Regarding Claim 53:**

**Kodosky et al. discloses** the medium of claim 52, wherein the function accepts at least one argument and returns at least one result. **(Page 1, Paragraph 9, Lines 1-4)**

**Regarding Claim 54:**

**Kodosky et al. discloses** the medium of claim 52, wherein the one or more transition components can invoke the function. **(Paragraph 132 Lines 1-5. Figure 8. Page 1, Paragraph 10, Lines 1-5)**

**Regarding Claim 55:**

**Kodosky et al. discloses** the medium of claim 52, further comprising instructions for accepting user input specifying data properties of the function. **(Page 1, Paragraph 9, Lines 7-9)**

**Regarding Claim 56:**

**Kodosky et al. discloses** the medium of claim 52, further comprising instructions for associating a data item with the function. **(Page 1, Paragraph 9, Lines 7-9. Page 2, Paragraph 11, Lines 2-7)**

**Regarding Claim 57:**

**Kodosky et al. discloses** the medium of claim 52, wherein the function comprises two or more graphical elements. **(Page 6, Paragraph 63, Lines 1-8)**

**Regarding Claim 58:**

**Kodosky et al. discloses** the medium of claim 52, wherein the function has a plurality of configurable properties. **(Page 1, Paragraph 10, Lines 1-5)**

**Regarding Claim 59:**

**Kodosky et al. discloses** the medium of claim 52, wherein the function defines a textual format for invoking the function. **(Page 12, Paragraph 132 Lines 1-5. Figure 8)**

**Regarding Claim 60:**

**Kodosky et al. discloses** the medium of claim 52, further comprising instructions for providing a shadowing function wherein shadowing comprises using in a function invocation a function definition proximally closest to a point of invocation of the function in a state diagram hierarchy. **(Page 3, Paragraph 20, Lines 8-13; Creators priority order can allow for closest function definition to execute.)**

**Regarding Claim 61:**

**Kodosky et al. discloses** A computer implemented method for modeling a system using a graphical block diagram environment, said method comprising:

graphically representing a function for use in an executable model within the graphical block diagram environment the executable model including at least one state or transition, the function being represented graphically and separately from the at least one state or transition in the executable model; and; **(Page 14, Paragraph 165, Lines 1-5. Figure 19)** and

textually referencing the graphically represented function within the model to cause an invocation of the graphically represented function during execution of the model. **(Paragraph 132 Lines 1-5. Figure 8)**

**Regarding Claim 62:**

**Kodosky et al. discloses** The computer implemented method of claim 61, wherein the model is represented as a finite state machine. **(Page 3, Paragraph 20, Lines 8-13)**

**Regarding Claim 63:**

**Kodosky et al. discloses** The computer implemented method of claim 62, wherein the finite state machine is a hierarchical finite state machine. **(Page 3, Paragraph 20, Lines 8-13)**

**Regarding Claim 64:**

**Kodosky et al. discloses** The computer implemented method of claim 62 further comprising:

Associating the graphically represented function with at least one state or transition within the finite state machine. **(Page 2, Paragraph 16, Lines 1-4)**

**Regarding Claim 65:**

**Kodosky et al. discloses** The computer implemented method of claim 61, wherein the graphically represented function is represented as at least one of a finite state machine, a state flow diagram, a function flow diagram, and a graphical block diagram model. **(Page 3, Paragraph 20, Lines 8-13)**

**Regarding Claim 66:**

**Kodosky et al. discloses** A medium holding instructions executable using the electronic device for modeling a system using a graphical block diagram environment, said instructions comprising instructions for:

Graphically defining a function for use in an executable model within the graphical block diagram environment the executable model including at least one state or transition, the function being represented graphically and separately from the at least one state or transition in the executable model; and **(Page 14, Paragraph 165, Lines 1-5. Figure 19)**

textually referencing the graphically represented function within the model to cause an invocation of the graphically represented function during execution of the model. **(Paragraph 132 Lines 1-5. Figure 8)**

**Regarding Claim 67:**

**Kodosky et al. discloses** The medium of claim 66, wherein the model is represented as a finite state machine. **(Page 3, Paragraph 20, Lines 8-13)**

**Regarding Claim 68:**

**Kodosky et al. discloses** The medium of claim 67 further comprising instructions for:

Associating the graphically represented function with at least one state of transition within the finite state machine. **(Page 2, Paragraph 16, Lines 1-4)**

**Regarding Claim 69:**

**Kodosky et al. discloses** The medium of claim 66, wherein the graphically represented function is represented as at least one or a combination of:

a finite state machine, **(Page 3, Paragraph 20, Lines 8-13)**

a state flow diagram,

a function flow diagram,

and a graphical block diagram model.

**Regarding Claim 70:**

**Kodosky et al. discloses** A computer implemented system for modeling using a graphical block diagram environment, said system comprising:

Means for representing a function defined graphically for use in an executable model within the graphical block diagram environment the executable model including at least one state or transition, the function being represented graphically and separately from the at least one state or transition in the executable model; and **(Page 14, Paragraph 165, Lines 1-5. Figure 19)**

Means for textually referencing the function defined graphically within the model to cause an invocation of the function during execution of the model. **(Paragraph 132 Lines 1-5. Figure 8)**

**Regarding Claim 71:**

**Kodosky et al. discloses** The system of claim 70, wherein the model is represented as a finite state machine. **(Page 3, Paragraph 20, Lines 8-13)**

**Regarding Claim 72:**

**Kodosky et al. discloses** The system of claim 71 further comprising means for associating the graphically represented function with at least one state of transition within the finite state machine. **(Page 3, Paragraph 20, Lines 8-13)**

**Regarding Claim 73:**

**Kodosky et al. discloses** The system of claim 70, wherein the graphically represented function is represented as at least one or a combination of a finite state machine, a state flow diagram, a function flow diagram, and a graphical block diagram model. **(Page 3, Paragraph 20, Lines 8-13)**

**Regarding Claim 74:**

**Kodosky et al. discloses** A graphical block diagram modeling system comprising:

A graphical function for use in an executable model, wherein at least subset of commands of the graphical function are defined through a graphical representation the executable model including at least one state or transition, the graphical function being represented graphically and separately from the at least one state or transition in the executable model; and; **(Page 14, Paragraph 165, Lines 1-5. Figure 19)** and

A graphical representation of the model including a textual reference of the graphically represented function within the graphical representation of the model to cause an invocation of the graphical function during execution of the model. **(Paragraph 132 Lines 1-5. Figure 8)**

**Regarding Claim 75:**

**Kodosky et al. discloses** The system of claim 74, wherein the model is represented as a finite state machine. **(Page 3, Paragraph 20, Lines 8-13)**

**Regarding Claim 76:**

**Kodosky et al. discloses** The system of claim 75, wherein the finite state machine is a hierarchical finite state machine. **(Page 3, Paragraph 20, Lines 8-13)**

**Regarding Claim 77:**

**Kodosky et al. discloses** The system of claim 75, wherein the finite state machine further comprises at least one state or transition associated with the graphical function. **(Page 3, Paragraph 20, Lines 8-13)**

**Regarding Claim 78:**

**Kodosky et al. discloses** The system of claim 74, wherein the graphical function is represented as at least one or a combination of:

a finite state machine, **(Page 3, Paragraph 20, Lines 8-13)**

a state flow diagram,

a function flow diagram,

and a graphical block diagram model.

***Conclusion***

**4.** All Claims are rejected.

**5.** Any inquiry concerning this communication or earlier communications from the examiner should be directed to Saif A. Alhija whose telephone number is (571) 272-8635. The examiner can normally be reached on M-F, 11:00-7:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kamini Shah can be reached on (571) 272-22792279. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

KAMINI SHAH
SUPERVISORY PATENT EXAMINER

SAA

November 23, 2007